

# Programming Sensor Networks with Nomadic NFC Transponders

Teemu Leppänen, Joonas Kataja, Iván Sánchez Milara and Jukka Riekkö

Center for Ubiquitous Computing, University of Oulu, Finland

{teemu.leppanen, joonas.kataja, ivan.sanchez, jukka.riekki}@ee.oulu.fi

**Abstract**— We present how NFC transponders can be used for energy efficient programming of closed-loop sensor networks, to update or augment the existing functionality. Use cases include road tunnel inspection, water pipeline monitoring and maintaining safety information on behalf of mine workers. We utilize opportunistic movement of the human operator, the flow of fluid in a pipeline or material in mines, to move the NFC transponder in the system effortlessly and without external network connectivity. Transponders contain mobile agents in their memory, which are injected into the system when transponder comes to the proximity of a node with NFC reader component. Then mobile agents autonomously operate their tasks, i.e. collect and process sensor data in the devices, detect events from data, control physical components and report their results. Mobile agents can adapt to the operational conditions of the system and physical environment, e.g. to save energy or operate in isolated network segments in fault situations. Real-world evaluation shows that this method is energy efficient in comparison with communications atop similar wireless sensor network.

**Keywords**—Sensor networks, Near Field Communication, Mobile agent, Road tunnel, Water pipeline, Mine safety

## I. INTRODUCTION

Wired and wireless sensor networks (WSN) have emerged as highly flexible distributed application platforms to observe practically all types of urban, rural and industrial environments. Sensor networks (SN) also produce data about the system behavior for closed-loop control systems, which have tighter real-time requirements for data collection and processing.

WSN relies on wireless communications to transmit data from collecting sensor nodes through gateways to backend systems. The benefit of WSN is that nodes could be placed anywhere in environment, where fixed network infrastructure could be a costly option, such as road tunnels [1]. However, wireless communications are not feasible for all types of SN deployments, including for example underground SN used for water pipeline monitoring [2], due to the physical characteristics of the environment. Moreover, uploading data to the backend system for analysis introduces significant latencies and energy consumption in nodes. The requirement for gateway devices complicates the deployment, increases the costs of operation and maintenance and adds significantly to the system energy consumption.

Although SN have been utilized in a large number of applications, programming the SN functionality still poses challenges [3]. The nodes are typically resource-constrained embedded devices with limited processing

power, limited in-device memory and have limited communication capabilities. Therefore the node firmware can't be preloaded with a great deal of functionality for data collection, analysis, component control and system monitoring tasks. Typically, the nodes are programmed once before deployment and further updating or augmenting the functionality is challenging if possible at all. Methods such as over-the-air programming distribute software updates to the nodes through wireless communications and system infrastructure services, which are not easily available in all types of SN deployments.

In this work, we present a method for programming closed-loop SN and WSN deployments utilizing Near Field Communication (NFC) transponders, i.e. tags. We introduce the term *nomadic tag* to describe NFC tag that opportunistically physically moves in a system while storing a mobile agent in its memory. Opportunistic tag movement is enabled through inherent fluid or material flow in the system or by intentionally bringing the tag to the environment by a human operator, such as road tunnel inspection crew. This enables mobile agents to operate in the system in the absence of communication infrastructure. Once the tag appears in proximity of a NFC reader node, it migrates into the system through read operation. The mobile agent then performs different tasks in the system: collects and processes sensor data, detects and reports events, controls physical components and monitors the system state.

This method minimizes the SN programming effort and is an energy efficient alternative in comparison with other programming methods that require communication infrastructure and system services. The contribution of this paper is to show that this method is beneficial in several ways for tunnel, pipeline and mine SNs programming. In these use cases, the infrastructure is not easily available in large-scale, e.g. physically long, closed-loop sensor and control system deployments. Solutions exist to transmit mobile agents in Radio Frequency Identification (RFID) tags in distributed applications [4], but relying on network connectivity and infrastructure services to compile the agents from high-level instructions stored in the tag memory and where the agents are executed in resource-rich device platforms. As shown in this paper, nomadic tags are cheap and low-energy alternative that enable mobile agent operations in resource-constrained SN, as a tiny self-containing computational task unit.

The rest of the paper is organized as follows. In Section 2, we describe the method in detail. Section 3 discusses the benefits for the presented use cases. In Section 4, we evaluate the overhead generated by this method in comparison with similar WSN operations. Related work is presented in Section 5 and Section 6 concludes the paper.

## II. PROGRAMMING SENSOR NETWORKS

As discussed in detail in Section III, tunnel and pipeline SN's are typically sense-only applications [3], where sensor data collected by nodes is sent further to the centralized backend system for analysis. After analysis, in some cases, control commands are sent back to SN. Data flow is based on many-to-one communications. On the other hand, sense-and-react applications [3] utilize continuous sensing and decentralized in-network data processing in the deployment location that allows taking local system properties and environmental conditions into consideration to optimize the SN operations. In-network processing reduces communication latencies and system energy consumption, as the amount of transmitted data is reduced, which also increases system reliability. In-network processing assists in providing real-time response to detected events or faults and enables operations on an isolated segment of the SN. This is an important feature for safety critical applications, such as road tunnel operation and mine safety.

We propose how mobile agents can be used to implement in-network data processing in tunnel, pipeline and mine SNs. Mobile agents are used to deliver application-specific tasks into the SN to update node firmware, introduce new functionality into the system and optimize system operation. Updated or new utility data can be delivered with mobile agents to the closed-loop SN without external network connectivity to optimize node operations, e.g. data for sensor calibration or event detection. Once injected into the system, mobile agents then operate autonomously to execute their tasks. Another advantage of mobile agents is to abstract device heterogeneity, tasks can then be addressed to different types of SN nodes with different sensors and to consider their individual capabilities in task execution. Mobile agents have been shown to reduce communication costs and the amount of transmitted data in WSN [5].

Furthermore, we enable sense-and-react applications in these types of SNs by utilizing nomadic NFC tags as the physical carrier to transmit mobile agents. The tags are nomadic in a sense that their physical movement in systems is facilitated by inherent system properties, e.g. worker movement, fluid flow in a pipeline over long distances or material flow in mines. This method requires little effort of the system operators, and system components can operate without expensive infrastructure. The close physical proximity requirement of the tag to the nodes makes this method less vulnerable to attacks than infrastructure-based methods.

The tasks of mobile agents in sense-and-react applications include, first, to process the sensor data to calculate application-specific data features and detect events. Mobile agents utilize several types of data in their tasks: control data stored into the tag to execute actions, tag stored utility data and sensor data describing the environment. We include this data into the mobile agent composition, thus the agent becomes a self-contained unit for these tasks. After task execution, the agent stores its result and updates its data. Secondly, the processed data, events or agents' results are transferred with the agent in the system or communicated using infrastructure if available. Here, mobile agents can implement application-specific many-to-many communications for SN applications. However, due to the resource-constrained

nature of the SN nodes, lightweight communication protocols and data representation models are to be considered [6]. Thirdly, mobile agents control system behavior or modify its state by actuating physical components that are attached to nodes.

Due to the mobile agent's autonomous migration capability, their tasks and data can migrate between nodes, to where it is the most appropriate to execute the tasks. Mobile agents can consider the local system environment characteristics to adapt and optimize their task execution, from the point-of-view of an individual node but also of the whole network or the application. These are important considerations for WSN programming [3], as in centralized SN approaches these features are difficult to implement and are prone to connection or node failures in runtime. Mobile agents can be implemented to perform tasks in local, group or global scale, depending on the resources the mobile agent uses and where it migrates. Mobile agents can be programmed to operate only in specific segments of the network.

The simplest mobile agent task would be to operate as a data collecting ferry [7], that travels across the network to collect data from the nodes and deliver it to the backend system. Ferries can also be used to transfer data between isolated SN segments. Benefits are the reduced communication costs for the nodes, when the application can tolerate delays in data delivery. A solution to the delay problem could be to inject a number of mobile agents into the system that operate simultaneously in parallel. Similarly, on-demand tasks can be implemented that are injected into the system by a human operator or started based on detected events.

The only infrastructure requirement for this method is to utilize gateway nodes in the SN that are equipped with NFC tag reader components. Also, a physical mechanism, such as a net, needs to be attached to the node and controlled to hold a nomadic tag in the reading distance long enough for read or write operations. When the nomadic tag is introduced to the physical proximity of the reader node, it enables the agent to migrate into the node by NFC read operation. The node then executes the agent and compiles an agent message to migrate it into the next node. After the agent's task is completed in the SN, it migrates to the gateway node. The node then enables further migration to the external systems by writing the agent back to passing by nomadic tag or to the same tag if its movement is delayed by the reader node.

### A. Enabling technologies

RFID technology uses radio waves transmission to communicate with a wide variety of physical objects. RFID system consists of transponders, i.e. tags, attached to the objects and reader components to connect the system into external networks. The reading range varies from several centimeters to a few tenths of meter depending on the RFID technology. Tags are composed of a microchip, that can store and process data with varying memory sizes, and an antenna. Tags are categorized into active, that require its own power source, and passive, that receive power from the reader's electromagnetic field. The typical use case for RFID technology is to store an object identifier into the tag memory that is used to retrieve the object specific data from a system information repository. In code-centric RFID systems, the tag information is used to retrieve the instructions for actions to be executed in

the system from the repository [4], [8]. Near Field Communication (NFC) is RFID based technology that provides low power direct communication among NFC-enabled devices. It exhibits reading the distance of a few centimeters. NFC Data Exchange Format (NDEF) is a light-weight binary format to store data into the tag memory. A NDEF record consists of multiple records as an array, which have predefined type, such as text, URI, MIME type or external type. For security, tags contain unique identifier and support data encryption. Tags also feature key-based access control for read and write operations and for different elements separately, providing multiple levels of security.

### B. Mobile agent considerations

To create a mobile agent, it is first needed to define and implement its task as a program. Secondly, the resources that are required for the task execution, i.e. sensor or data types, need to be defined. The resources can also include external utility data that is used in the task, e.g. sensor calibration coefficients or event detection thresholds. The mobile agent architecture used in this work is presented in [6]. The task code is precompiled to the node’s native machine language, which simplifies the programming effort for the resource-constrained nodes as the code can be written into the program memory as it is without language interpretation step, even though the architecture enables to use higher-level scripting languages in the agent task code. Thus the mobile agent includes the task code as precompiled binary. The resources in nodes are described and accessed through URLs that contain the node address and resource identifier, e.g. “temperature” represents a temperature sensor in the node. When the mobile agent migrates into a node, it retrieves the resource values through the interface provided by the node [6].

In this work, for SN communication we utilize the Constrained Application Protocol (CoAP)<sup>1</sup> that features minimal protocol overhead for communication through the Internet. The utilization of CoAP enables seamless integration between disparate systems, reduces the resources needed for protocol translation and message interpretation in the nodes. As CoAP has small protocol overhead, it makes this method feasible for constrained SN nodes and for low-end NFC tags with little memory. Therefore, the mobile agent is encapsulated in a binary CoAP message, with specific mobile agents related message options as described in [6]. Now, entire mobile agent as a self-contained unit can be reduced to a few hundred bytes, as shown later in this paper. Finally, the CoAP message is written into the NFC tag as a NDEF external record. There are two possible ways to store the agent: either as a single record, in case the agent is encapsulated in a single message, or (2) multiple records, when each record stores different mobile agent element, such as code and resources. The second method enables to modify agent elements separately, but require additional interpretation step in the nodes. Several mobile agents or other communication messages can be stored in the tag memory as its size permits. The key-based authentication mechanism in NFC tags is used for read and write operations and to access different SN resources.

The use of CoAP as the WSN communication protocol enables to define system-wide uniform interface and

communication primitives for the WSN and for the mobile agent operations [6]. We follow the RESTful approach presented in [6], where CoAP methods are used to retrieve sensor data or mobile agent task results from a node (GET), inject a mobile agent into a node, send an event or actuate a physical component (POST), and remove an agent from the system (DELETE).

## III. USE CASES

We present three use cases for SN programming with nomadic tags that benefit from minimized programming effort and reduced infrastructure deployment costs. Nomadic tags provide unlimited migration and communication distance with no extra energy consumption. However, some use case-specific latency is introduced in the tag movement, i.e. the mobile agent migration is determined by the physical properties of the environment.

### A. Road tunnels

The inspection of road tunnels routinely requires human operators or robots [9], who identify problems visually or by using specific equipment. A concern is the economic impact of closing the tunnel for inspection. Other concerns include energy consumption optimization, the quality and timeliness of monitoring data, which have utmost importance for traffic flow and safety. SN deployments with different types of sensors assist in collecting data about the tunnel operation and environmental conditions. That data is sent to a control station that processes the large amount of data and reacts to the results, if needed [10]. However, in large-scale deployments, such as long tunnels, the operating conditions and data processing requirements at the locations of nodes are different in separate parts of the tunnel [10]. For example, at the entrance of a tunnel the ambient light conditions are different than further in the tunnel. The data collected by sensors in these deployments may not provide enough information for a complete assessment of the tunnel condition [9]. Thus, node and operating environment heterogeneity would benefit from decentralized data processing [11]. Moreover, wireless communications in tunnel WSN require specific networking techniques due to the environment [1].

Therefore, local environmental conditions, nodes’ communication capabilities and energy efficiency are to be taken into account in the SN operation and in control of physical components in tunnels, such as lighting. Mobile agents assist in reducing the resource consumption of both individual nodes and the whole SN. Agents autonomously adapt to the local conditions, i.e. improve the node energy consumption optimizing sensing parameters, such as utilized sensors, sample rates and sleep modes, in relation to observed system behavior, data and events. Mobile agents communicate with each other to aggregate their data and distribute task computation and communication load by migration. Mobile agents operate autonomously in isolated SN segments, despite of connection failures, malfunctioning nodes or changes in the operating environment.

The tunnel SN programming operation can be conducted as follows. The inspection or maintenance crew carries their nomadic tags, with mobile agents stored in their memory, to the tunnel and inject agents into the SN

<sup>1</sup> <https://datatracker.ietf.org/wg/core/charter/>

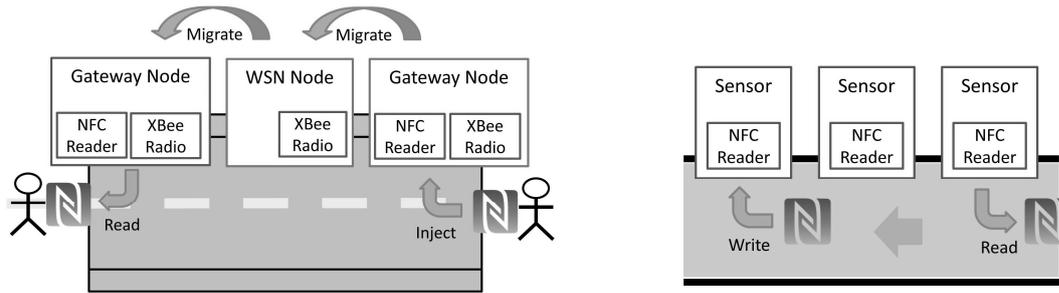


Figure 1. Nomadic tag mobile agent operations in the presented use cases: road tunnel (left) and water pipeline (right).

by passing by the gateway node equipped with NFC reader. Then mobile agents operate in the tunnel SN until their tasks are complete. The results of the agent operations can be read from a gateway node at the other end of the tunnel, once the agents migrate into that node. During the operations, visual cues can be given to the crew about system state, in case the nodes or tunnel equipment can be actuated by the mobile agent.

See Fig. 1 for the illustration of the method in tunnel WSN. We evaluated the energy efficiency and latencies of this method in Section IV, showing that NFC tag read and write operations are in range of one second. This latency does not significantly interrupt the flow of traffic in a road tunnel, e.g. when passing by the gateway node with a vehicle.

Related work on road tunnel WSN programming includes separating the deployment into the logical subsets of nodes that implement decentralized data processing, based on the common node characteristics [11]. Otherwise, WSN architecture to adaptively control sensing parameters in tunnel conditions was presented in [10]. WSN gateways upload sensor data from nodes to a centralized controller that analyses the data, changes node configuration parameters and controls the lighting of a tunnel in real-time.

### B. Pipeline monitoring

Both wired and wireless sensor networks have been utilized in continuous water (and sewer) pipeline monitoring and maintenance [2], [12]. Challenges include cost-effectiveness as dynamic proactive monitoring is required due to large economic loss, if a fault occurs. Typically, data is sent from densely deployed nodes outside or inside a pipeline through a gateway to a control station for analysis. Deployments are challenging as the pipelines are often underground where providing power for the sensor network, installation of monitoring system and repairing cabling faults are difficult. Moreover, the size and scale of pipeline installations are increasing, therefore monitoring requires geographically large SN deployment, in which installation and maintenance are costly. Mobile sensor nodes, i.e. sensor nodes that move with inherent fluid flow in the pipeline, are useful for pipeline monitoring for several reasons [13]: mobile sensors provide better localization accuracy of detected faults, no energy is required for the flow-based movement and on demand monitoring of the pipeline becomes possible. Typically, static beacons, attached inside or outside of the pipeline receive data from the mobile nodes, provide location information for the data and disseminate it further. Challenges include optimizing where, when and

how many mobile nodes to insert into the pipeline. Due to mobility, increased data transmission power in the mobile nodes is a concern [13].

With nomadic tags and mobile agents, we are able to address several of the above-mentioned challenges. Nomadic tags provide cost-efficient and scalable way of inserting monitoring and controlling tasks into the pipeline. Mobile agents, as data ferries, enable opportunistic on-demand data transmission and communication between nodes. Data processing in the nodes enables monitoring the state of the pipeline, event detection and actuation of physical components, such as valves. Mobile agents are customizable and several mobile agents in tags can be easily injected into the pipeline for different tasks. For example, a set of writer tags can be inserted into the pipeline, and later, another set of reader tags follow their route. See Fig. 1 for the illustration of this method. Efficient latching mechanism, such as net, is needed in the nodes to hold the tag in the reader range long enough for read and write operations. To minimize missed opportunities in tag reading, several tags with the same task code can be released into the pipeline periodically. For some tags, waterproof casing may be needed. We evaluated the energy efficiency and latencies of this method in Section IV.

Related work includes WSN deployments with resource-rich sensors at pipeline checkpoints which also performs in-network data processing before sending the data further to the backend system [14]. The backend can then notify checkpoint sensors to collaboratively locate the detected faults. Waterproof mobile nodes are released into a water pipeline in [15]. Mobile nodes are used to collect data, disseminate it between themselves and send it to static beacons outside the pipeline in [13]. The nodes travel in the pipeline, locating their target positions and then latch themselves into the pipeline surface. Once node's battery has run out, it unlatches itself and exits the pipeline with the water flow. Passive RFID tags are used in [12] as a localization and event data storage method, cooperatively with mobile sensors, which reduces the memory requirement for the mobile nodes. The tags are powered by RFID reader in the mobile node to read and write data from / to the tag, and mobile node operate as a ferry to take the data out of the pipeline. RFID tags, both active and passive, are attached to pipelines to provide location information and data storage, where mobile sensor nodes equipped with readers would float along the pipeline in [16][17].

### C. Mine safety

The challenges described previously for tunnel and pipeline SN largely also concern SN deployments in mines [18]. A mine safety monitoring threats include personnel safety, machinery condition, the detection of accidents and landscape changes [18]. Here, we have two ways to move mobile agents inside the mine. Firstly, nomadic tags can be attached to the equipment or material, thus the agent would move along its physical movement. Secondly, personalized mobile agents of the workers, stored in nomadic tags, would be injected into the mine SN when the workers enter the mine. We assume readers would be installed along the mine.

An example task of the mobile agents would be to keep track of the worker location at all times and monitor the immediate environment at the location for the signs of danger, through processing the sensor data. Mobile agents would react to detected events, received event notifications and to the mine SN faults. Another use case would be, in case of an emergency, to try to communicate with the outside world on behalf of the worker, to report their locations and even try to find safe evacuation routes inside the mine by migration in the functioning SN segments. However, these use cases require that the information provided by the agent should be visualized by the trapped workers with their personal devices.

Related work includes using active RFID devices attached to workers or equipment to create mesh networks [19]. Use cases include monitoring environment with attached sensors, locating workers and assets in the mine and preventing accidents.

## IV. EVALUATION

To evaluate this method in real-world experiment, we implemented a prototype WSN that was accessed with a nomadic tag, first to program the nodes and second to collect data from nodes. We stored the mobile agent as a CoAP message [6] into the tag as a single NDEF record. As the sensor nodes, we utilized Arduino Micro boards, based on Atmel’s 8-bit microcontrollers, with 2.5 kB of available RAM. The gateway nodes were Arduino Mega boards with attached NFC reader (PN532). All nodes communicated with XBee radios on 2.4 GHz frequency band. We injected a NFC tag with a mobile agent into a plastic water pipeline (diameter 10 cm, length 4 m) by hand and let it move along with the water flow. The flow was created by a tap attached to the pipeline. We observe that reading the tag data inside a pipeline, with the reader positioned outside surface of the pipeline, succeeded through water from 5 cm distance.

We utilized MIFARE Classic 1K as the nomadic NFC tag with basic key-based security that leaves 752 bytes of available memory to store the mobile agent. According to our results, shown in Table 1, this tag memory size would allow a single mobile agent operation in a SN consisting of 20 nodes. In the experiments, the agent size in bytes was less than 250 bytes with one resource, e.g. node address or sensor data type. The mobile agent’s task was to migrate into nodes from the nomadic tag to read sensor data of specific sensor type, as defined as the resource, and return data as its result to the tag. The task code was written in the C programming language and precompiled into nodes’ native machine language format, as described in [6]. More complex tasks, that utilize more resources,

TABLE I.  
MOBILE AGENT COAP MESSAGE SIZE

Element	Size (B)	Addition (B) / Node
CoAP Protocol overhead	21	10
Task code	183	20
Node addresses	-	10
Result size	5	5

TABLE II.  
LATENCIES OF MOBILE AGENT OPERATION VS. WSN OPERATION (N=3)

	Operation	Average Latency (ms)
<b>Mobile agent operations</b>		
NFC reader	Detect tag by polling	26
	Read CoAP message	331
	Write CoAP message	383
	Addition / Node ( Read / Write)	34 / 12
Gateway / WSN Node	Parse CoAP message, execute mobile agent, update result and compile new CoAP message	141
<b>WSN operation</b>		
XBee radio	Send CoAP message and wait for acknowledge	336

can be implemented with existing larger size NFC tags with kilobytes of memory.

Latencies in mobile agent operations, in comparison with similar WSN operations, are shown in Table 2. We observe that mobile agent migration from NFC reader adds some latency according to the number of utilized resources in nodes, in comparison with sending the CoAP message over a wireless link. The operational latencies are in range from 500ms to 1s with an agent operating in WSN size of 20 nodes. The manufacturer reported power consumption for PN532 is 500mW in continuous mode. For the XBee radios, the reported consumption is around 40-295mW. However, the NFC reader components are used infrequently in the presented use cases.

## V. RELATED WORK

Previous work demonstrated RFID technology used with WSN to report sensor-related activity and to collect sensor data [8]. Also, RFID technology provides means for node localization in SN deployments [2]. RFID technology has been used with mobile agents before. To execute actions and to represent new service functionality in code-centric RFID systems, mobile agents transmitting code in tag memory have been utilized in [4], [20], [21]. These deployments include system information repositories that are set up in advance. In case of long distance SN deployments, these repositories would have to be replicated in the communication range of each node that adds significantly to the infrastructure costs and introduces communication overhead to the system operation. Using application-specific high-level programming languages to describe the mobile agent task, as in the Wiseman architecture [4], requires additional language interpretation step in a code parser component in the system, before the actions can be executed in the devices. The implementations in [20], [21] are based on

the Java-based mobile agents JADE framework, in which case the agent size is too large (starting from tens of kilobytes) to be included in NFC tags and requires running a Java virtual machine in the node platform. This excludes the resource-constrained embedded networked devices to execute mobile agents. In [20], mobile agent-based software framework utilizing NFC tags was presented that utilizes infrastructure services for its operation. NFC tag stores the mobile agents and ascertained object ID's that are used by a broker component to retrieve agent information from the system repository. A dispatcher component then launches the agent, monitors and tracks its execution. A-FRED [21] utilizes XML to encode agent behaviors as rules and object data onto RFID tags. A-FRED agent then decodes the information from the tag and passes it to a system agent. The system agent maps the agent behavior rules against a pre-deployed schema file in the platform and creates defined worker agents (i.e. mobile agents) to execute the tasks in the specified platforms. The worker agents are able to migrate, but are cannot coordinate their tasks autonomously.

## VI. DISCUSSION

We have presented how nomadic NFC tags can be used for mobile agent-based programming of closed-loop SN deployments without external communication capabilities. Mobile agents collect and process sensor data, detect events, control physical components and disseminate task results, data and events in the system. In their operation, mobile agents autonomously adapt to the changing environment and can continue operations in isolated network segments in fault situations. Example use cases include road tunnel inspection and monitoring the tunnel and tunnel SN operations. Also, we presented water pipeline monitoring SN operations, based on mobile agents in nomadic tags moving effortlessly with the fluid flow. Yet another use case is maintaining mine safety information, by analyzing sensor data, for workers while they are working in the mine. Our real-world evaluation shows that this method is energy efficient in comparison with communications atop similar WSN. Our future work includes implementing more complex scenarios of WSN programming and adding more intelligence to the mobile agents.

## REFERENCES

- [1] L. Mottola, G. Picco, M. Ceriotti, S. Guna, and A. Murphy, “Not all wireless sensor networks are created equal,” *ACM Transactions on Sensor Networks*, vol. 7, no. 2, pp. 1–33, August 2010.
- [2] M. BenSaleh, S. Qasim, A. Obeid, and A. Garcia-Ortiz, “A review on wireless sensor network for water pipeline monitoring applications,” In: *International Conference on Collaboration Technologies and Systems*, pp. 128–131, San Diego, CA, May 20-24, 2013.
- [3] L. Mottola and G. Picco, “Programming Wireless Sensor Networks: Fundamental Concepts and State of the Art,” *ACM Computing Surveys*, vol. 43, no. 3, pp. 1–51, April 2011.
- [4] M. Chen, S. González, Q. Zhang, and V. Leung, “Code-centric RFID system based on software agent intelligence,” in *IEEE Intelligent Systems*, vol. 25, no. 2, pp. 12–19, March-April 2010.
- [5] M. Chen, T. Kwon, Y. Yuan, and V. Leung, “Mobile Agent Based Wireless Sensor Networks,” *Journal of Computers*, vol. 1, no. 1, pp. 14-21, 2006.
- [6] T. Leppänen, M. Liu, E. Harjula, A. Ramalingam, J. Ylioja, P. Närhi, et al. “Mobile Agents for Integration of Internet of Things and Wireless Sensor Networks,” In: *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 14–21, Manchester, UK, 13-16 October, 2013.
- [7] M. Alnuaimi, K. Shuaib, K. Alnuaimi, and M. Abdel-Hafez, “Data Gathering in Delay Tolerant Wireless Sensor Networks Using a Ferry,” *Sensors*, vol. 15, no. 10, pp. 25809–25830, 2015.
- [8] R. Want, “Enabling ubiquitous sensing with RFID,” *Computer*, vol. 37, no. 4, pp. 84–86, April 2004.
- [9] L. Mottola, G. Pietro Picco, M. Ceriotti, Ş. Gună, and A. L. Murphy, “Not all wireless sensor networks are created equal,” *ACM Transactions on Sensor Networks*, vol. 7, pp. 1–33, 2010.
- [10] C. Balaguer, R. Montero, J. Victores, S. Martinez, and A. Jardón, “Towards Fully Automated Tunnel Inspection: A Survey and Future Trends,” In: *31st International Symposium on Automation and Robotics in Construction & Mining*, pp. 19-33, Sydney, Australia, 9-11 July, 2014.
- [11] M. Ceriotti, M. Corra, L. D’Orazio, R. Doriguzzi, D. Facchin, S. Guna, et al., “Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels,” In: *10th International Conference on Information Processing in Sensor Networks*, pp. 187–198, Chicago, IL, 12-14 April, 2011.
- [12] L. Mottola and G. Picco, “Programming wireless sensor networks with logical neighborhoods: a road tunnel use case,” In: *5th ACM Conference on Embedded Networked Sensor Systems*, pp. 393–394, Sydney, Australia, 6-9 November, 2007.
- [13] J. Kim, G. Sharma, N. Boudriga, and S. Iyengar, “SPAMMS: A sensor-based pipeline autonomous monitoring and maintenance system,” in *2nd International Conference on Communication Systems and Networks*, pp. 1-10, Bangalore, India, 5-9 January, 2010.
- [14] M. Suresh, W. Zhang, W. Gong, R. Stoleru, A. Rasekh, and M. Banks, “Toward Optimal Monitoring of Flow-Based Systems Using Mobile Wireless Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 11, no. 3, article 48, May 2015.
- [15] Z. Sun, P. Wang, M. Vuran, M. Al-Rodhaan, A. Al-Dhelaan, and I. Akyildiz, “MISE-PIPE: Magnetic induction-based wireless sensor networks for underground pipeline monitoring,” *Ad Hoc Networks*, vol. 9, no. 3, pp. 218–227, May 2011.
- [16] T. Lai, W.-J. Chen, K.-H. Li, P. Huang, and H.-H. Chu, “TriopusNet: automating wireless sensor network deployment and replacement in pipeline monitoring,” In: *11th International Conference on Information Processing in Sensor Networks*, pp. 61–72, Beijing, China, April 16-19, 2012.
- [17] S. Rekhis, N. Ellouze, and N. Boudriga, “A Wireless Sensor Network Based Water Monitoring System,” In: *8th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 33–40, Paphos, Cyprus, October 21-25, 2012.
- [18] A. Almazyad, Y. Seddiq, A. Alotaibi, A. Al-Nasheri, M. BenSaled, A. Obeid, et al., “A Proposed Scalable Design and Simulation of Wireless Sensor Network-Based Long-Distance Water Pipeline Leakage Monitoring System,” *Sensors*, vol. 14, no. 2, pp. 3557–3577, February 2014.
- [19] N. Reddy, M. Saketh, and S. Dhar, “Review of sensor technology for mine safety monitoring systems: A holistic approach,” In: *IEEE First International Conference on Control, Measurement and Instrumentation*, pp. 429–434, Kolkata, India, 8-10 January, 2016.
- [20] L. Bandyopadhyay, S. Chaulya, P. Mishra, A. Choure and B. Baveja, “Wireless Information and Safety System for Mines,” *Journal of Scientific and Industrial Research*, vol. 68, pp. 107-117, February 2010.
- [21] Y. Chang, C. Fan, and Y. Wu, “Agent-based Intelligent software exploits near-field communication,” *IT Professional*, vol. 13, no. 2, pp. 30-36, March-April 2011.
- [22] C. Chan, H. Chow, W. Siu, H. Ng, T. Chu, H. Chan, et al., “A Multi-Agent-based RFID Framework for Smart-object Applications,” In: *International MultiConference on Engineers and Computer Scientists*, vol. 1, pp. 12–17, Hong Kong, March 14-16, 2012.